INTERNET-DRAFT
November 17, 2003
Expires: May 17, 2004

John Lazzaro
CS Division
UC Berkeley

Framing RTP and RTCP Packets over Connection-Oriented Transport

<draft-ietf-avt-rtp-framing-contrans-00.txt>

Status of this Memo

Copyright Notice

Abstract

This memo defines a method for framing Real Time Protocol (RTP) and Real Time Control Protocol (RTCP) packets onto connection-oriented transport (such as TCP and TLS). The memo also defines how to specify the framing method in a session description.

Table of Contents

1. Introduction

The Audio/Video Profile (AVP, [1]) for the Real-Time Protocol (RTP, [2]) does not define a method for framing RTP and Real Time Control Protocol (RTCP) packets onto connection-oriented transport protocols (such as TCP and TLS). However, earlier versions of RTP/AVP did define a framing method, and this method is in use in several implementations.

In this memo, we document the method and show how a session description [4] may specify the use of the method.

1.1 Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14, RFC 2119 [11].

2. The Framing Method

Figure 1 defines the framing method.

```
  0                   1                   2                   3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
  ---------------------------------------------------------------
 |             LENGTH             |  RTP or RTCP packet ...       |
  ---------------------------------------------------------------
```

Figure 1 -- The bitfield definition of the framing method.

A 16-bit unsigned integer LENGTH field, coded in network byte order (big-endian), begins the frame. If LENGTH is non-zero, an RTP or RTCP packet follows the LENGTH field. The value coded in the LENGTH field MUST equal the number of octets in the RTP or RTCP packet. Zero is a valid value for LENGTH, and codes the null packet.

This framing method does not use frame markers (i.e. an octet of constant value that would precede the LENGTH field). Frame markers are useful for detecting errors in the LENGTH field. In lieu of a frame marker, receivers SHOULD monitor the RTP and RTCP header fields whose values are predictable (for example, the RTP version number).

3. Undefined Properties

The framing method does not specify properties above the level of a single packet. In particular, Section 2 does not specify:

The number of RTP or RTCP streams on the connection.

> The framing method is commonly used for sending a single RTP or RTCP stream over a connection. However, Section 2 does not define this common use as normative, so that (for example) a memo that defines an RTP SSRC multiplexing protocol may use the framing method.

Bi-directional issues.

> Section 2 defines a framing method for use in one direction on a connection. The relationship between framed packets flowing in defined direction and in the reverse direction is not specified.

Packet loss and reordering.

The reliable nature of a connection does not imply that a framed RTP stream has a contiguous sequence number ordering. For example, if the connection is used to tunnel a UDP stream through a network middlebox that only passes TCP, the sequence numbers in the framed stream reflect any packet loss or reordering on the UDP portion of the end-to-end flow.

Out-of-band semantics.

Section 2 does not define the RTP or RTCP semantics for closing a TCP socket, or of any other "out of band" signal for the connection.

Memos that normatively include the framing method MAY specify these properties. For example, Section 4 of this memo specifies these properties for RTP sessions specified in session descriptions.

4. Session Descriptions for RTP/AVP over TCP or TLS

[3] defines how to specify connection-oriented media streams in session descriptions. In this section, we show how to use [3] with the framing method.

Figure 2 shows the syntax of a media (m=) line [4] of a session description:

```
      "m=" media SP port ["/" integer] SP proto 1*(SP fmt) CRLF
```

Figure 2 -- Syntax for an SDP media (m=) line (from [4]).

[4] defines "TCP" as the <proto> token that specifies TCP transport, and [3] defines "TLS" as the <proto> token that specifies TLS transport. We now define how to declare that an RTP/AVP stream that uses the framing method appears on the TCP or TLS connection.

At least two <fmt> tokens MUST follow <proto>. The first <fmt> token MUST be "RTP/AVP". Subsequent <fmt> tokens MUST be unique unsigned integers in the range 0 to 127, that specify an RTP payload type associated with the stream.

The TCP or TLS <port> on the media line exclusively receives RTP packets. If a media stream uses RTCP, a second connection exclusively

receives the RTCP packets. The port for the RTCP connection is chosen using the algorithms defined in [4] and in related documents.

The TCP or TLS connections MAY carry bi-directional traffic, following the semantics defined in [3]. Both directions of a connection MUST carry the same type of packets (RTP or RTCP). The packets MUST exclusively code the RTP or RTCP streams specified on the media line(s) associated with the connection.

The RTP stream MUST have an unbroken sequence number order. RTCP stream packets MUST appear as defined in [2], with no lost or re-ordered packets. IETF standards-track documents MAY loosen these restrictions on packet loss and packet ordering.

The out-of-band semantics for the connection MUST comply with [3].

5. Example

The session descriptions in Figure 3-4 define a TCP RTP/AVT session.

```
v=0
o=first 2520644554 2838152170 IN IP4 first.example.net
s=Example
t=0 0
c=IN IP4 192.0.2.105
m=audio 9 TCP RTP/AVP 11
a=direction:active
```

Figure 3 -- TCP session description for first participant.

```
v=0
o=second 2520644554 2838152170 IN IP4 second.example.net
s=Example
t=0 0
c=IN IP4 192.0.2.94
m=audio 16112 TCP RTP/AVP 10 11
a=direction:passive
```

Figure 4 -- TCP session description for second participant.

The session descriptions define two parties that participate in a connection-oriented RTP/AVP session. The first party (Figure 3) is capable of receiving stereo L16 streams (static payload type 11). The second party (Figure 4) is capable of receiving mono (static payload

type 10) or stereo L16 streams.

The direction attribute in Figure 3 specifies that the first party is "active" and initiates connections, and the direction attribute in Figure 4 specifies that the second party is "passive" and accepts connections [3].

The first party connects to the network address (192.0.2.94) and port (16112) of the second party. Once the connection is established, it is used bi-directionally: the first party sends framed RTP packets to the second party on one direction of the connection, and the second party sends framed RTP packets to the first party in the other direction of the connection.

The first party also initiates an RTCP TCP connection to port 16113 (16112 + 1, as defined in [4]) of the second party. Once the connection is established, the first party sends framed RTCP packets to the second party on one direction of the connection, and the second party sends framed RTCP packets to the first party in the other direction of the connection.

A. Acknowledgements

This memo, in part, documents discussions on the AVT mailing list about TCP and RTP. Thanks to all of the participants in these discussions.

B. Security Considerations

Attackers may send framed packets with large LENGTH values, to exploit security holes in applications. For example, a C implementation may declare a 1500-byte array as a stack variable, and use LENGTH as the bound on the loop that reads the framed packet into the array. This code would work fine for friendly applications that use Etherframe-sized RTP packets, but may be open to exploit by an attacker.

C. IANA Considerations

[4] defines the syntax of session description media lines. We reproduce this definition in Figure 2 of Section 4 of this memo.

[4] defines "TCP" as a token value for the <proto> field of media lines, and [3] defines "TLS" as a token value for the <proto> field of media lines. [3] and [4] permit other memos to define tokens for the <fmt> fields that follow "TCP" or "TLS" on a media line.

This memo defines <fmt> tokens for use the "TCP" and "TLS" tokens. At least two <fmt> tokens MUST follow <proto>. The first <fmt> token MUST be "RTP/AVP". Subsequent <fmt> tokens MUST be unique unsigned integers in the range 0-127. Section 4 of this memo specifies the semantics associated with the <fmt> tokens.

D. References

D.1 Normative References

[1] Schulzrinne, H., and S. Casner. "RTP Profile for Audio and Video Conferences with Minimal Control", work in progress, draft-ietf-avt-profile-new-13.txt.

[2] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson. "RTP: A transport protocol for real-time applications", work in progress, draft-ietf-avt-rtp-new-12.txt.

[3] Yon, D. "Connection-Oriented Media Transport in SDP", work in progress, draft-ietf-mmusic-sdp-comedia-05.txt.

[4] Handley, M., Jacobson, V., and C. Perkins. "SDP: Session Description Protocol", work in progress, draft-ietf-mmusic-sdp-new-12.txt.

[5] Bradner, S. "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

E. Author Address

John Lazzaro
UC Berkeley
CS Division
315 Soda Hall
Berkeley CA 94720-1776
Email: lazzaro@cs.berkeley.edu

F. Intellectual Property Rights Statement

The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF's

procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

G. Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.